Network Formation Schemes for Dynamic Multi-Radio, Multi-Hop Wireless Cellular Networks

JaeSheung Shin, HeeSook Choi, Patrick G. Traynor, Thomas F. La Porta
Networking and Security Research Center
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
E-Mail: {jsshin, hchoi, traynor, tlp} @cse.psu.edu

***ABSTRACT* - Multi-hop relaying in cellular networks can greatly increase capacity and performance by exploiting the best available links to a base station. We envision an environment in which relay networks are dynamically formed in different frequency bands in response to the degradation of network performance. Nodes experiencing poor service may use their agile radios to join one of the available, non-interfering relay networks. We propose and evaluate a set of algorithms used to form such relay networks on-demand. Each of the algorithms begins by designating the nodes best suited for acting as gateways between the relay and cellular networks. Each scheme then determines the order of route request initiations. These algorithms are evaluated for latency, signaling overhead and gateway load during the network formation process, and average path length and amount of link sharing in the resulting relay networks. The evaluation leads us to conclude that having nodes furthest from the BS initiate route discovery first is the best approach for reducing the formation overhead and building efficient relay networks. To our knowledge, we are the first to propose and evaluate algorithms for the on-demand formation of multi-hop relay networks.**

I. INTRODUCTION

Mobile nodes in traditional wireless cellular networks communicate through centralized base stations (BS) in a pre-defined spectrum. All nodes within a cell share common spectrum. To improve the performance of such cellular networks, several studies on wireless relay networks, also called hybrid wireless networks, have been undertaken [1-4]. These solutions leverage the presence of persistent resources to support relay networks, and therefore rely on the fact that relay networks are always in existence.

We consider a multi-hop wireless cellular network comprised of agile radios in which relay networks are *dynamically formed* when performance on the radio access network is degraded. The spectrum for each relay network is allocated dynamically. In this way, multiple non-interfering relay networks may operate in parallel through the use of agile radios.

Because of the dynamic nature of these systems, the continuous existence of relay networks cannot be guaranteed. The spectrum on which a relay operates may be leased for a limited time. When network performance improves, or the spectrum on which it is operating is reclaimed, a relay network is dissolved and all nodes operate using the cellular interface once again. This motivates the need for an explicit procedure for mobile nodes forming a relay network. Figure 1 shows an example in which several groups of nodes form relay networks to a BS.

In this paper, we propose five relay network formation algorithms. Each algorithm first determines which nodes are best suited for acting as a bridge between the relay network and the BS, and designates these nodes as gateway (GW) nodes. The algorithms then discover paths from the nodes in the relay network to the GW nodes. A modified version of AODV [5] is used as the ad hoc routing protocol for the establishment of paths.

We compare the performance of these algorithms in terms of several metrics indicating the *overhead* of the relay network formation and the *efficiency* of the resulting relay networks. For formation overhead we consider latency, signaling traffic generated, and load at the GW nodes during the formation process. Because relay networks are typically formed during congested periods, or when the network is experiencing poor performance, formation latency is critical. Signaling traffic generated indicates the degree of network congestion during the network formation. The processing load at the GW nodes is proportional to the traffic intensity of the 3G interface between the BS and the GW nodes during the formation process.

To evaluate the efficiency of the resulting relay networks, we consider the average length of the paths from the nodes in the relay network to the GW nodes, and the amount of link sharing in the relay network. These metrics are good indicators of the resulting network performance.

The evaluation leads us to conclude that having nodes furthest from the BS initiate route discovery first is the best approach for reducing the formation overhead. Schemes based upon this approach can build efficient relay networks.

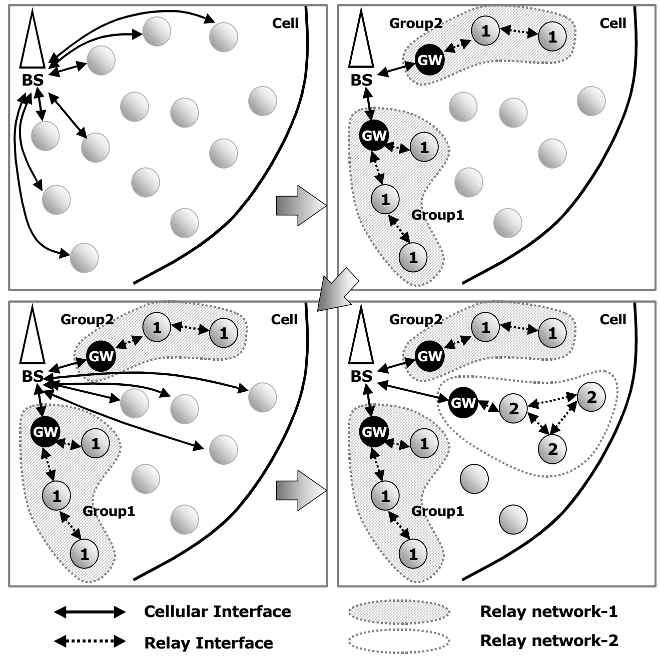


Fig. 1. Example of relay network formation

In this paper, we make the following contributions:

- We present a network model for a multi-hop wireless cellular network operating on multiple frequencies in which relay networks are dynamically formed and removed.
- We address the formation of relay networks for such an environment for the first time.

In addition to the network formation, the maintenance of a relay network is also important, but because it has been examined by others previously [1-4], we concentrate our focus on network formation.

The rest of the paper is organized as follows. In Section 2, we briefly discuss related work in the area. In Section 3, we present the architecture and properties of dynamic multi-frequency, multi-hop wireless cellular networks, basic operations used for relay network formation, and explain the network formation schemes in detail. In Section 4, we present our simulation environment and results. We discuss related issues in Section 5. Section 6 concludes this paper.

II. RELATED WORK

In this section we briefly review previous work on hybrid wireless (or relay) networks and network formation as applied to the algorithms presented here.

There has been a great deal of work on multi-hop wireless networks to improve cellular network performance [1-4]. Several papers address relaying in GSM networks [1][2]. We instead investigate a third generation (3G) wireless environment in which the sharing of communication resources is done via a combination of regulating power and time division multiplexing.

For example, in the 1xEV-DO system, the BS schedules only a single node for downlink transmission at any instant, and transmits at full power. The bit rate achieved during each time interval depends on its signal quality to the mobile node, which is a function of distance. If the BS can schedule nodes with better signal quality more often, then a higher average bit rate for the network can be achieved.

To maximize throughput of 3G networks, the UCAN [4] system proposes that the BS transmit all downlink data to mobile nodes with high signal quality, and that these nodes then forward data to other nodes in the network through a high speed relay network operating in a different spectrum than the 3G interface, specifically using an 802.11 network. In this way, the downlink from the BS can always run at its maximum rate and all users achieve higher throughput. It is reported that UCAN can achieve improvements of the average and maximum throughput of up to 82% and 37%, respectively. This work assumes that a persistent 802.11 network exists for use as the relay network and does not address the formation of relay networks under congested conditions.

The formation of Bluetooth networks [7-9] has also received significant attention. Because Bluetooth has no defined public broadcast channel, two Bluetooth devices cannot communicate with each other even when within radio range. They must, depending on geographical coverage, be first synchronized into piconets and then into larger scatternets. Routing protocols can then be run over the scatternet. The resulting formation algorithms are tightly coupled with special characteristics of Bluetooth networks. Accordingly, most research has focused only on reducing the amount of bridging overhead, the number of established Bluetooth links, and the number of piconets for minimizing inter-piconet interference. These algorithms are therefore not directly applicable to our environment.

To our best knowledge, there has been no research to deal with dynamic multi-hop wireless cellular networks operating on multiple frequencies. Moreover, there is no research to address the formation issue for this kind of network.

III. RELAY NETWORK FORMATION

In this paper, we consider an environment in which relay networks are formed when performance on the radio access network is degraded. If the BS schedules a large number of nodes that have poor signal quality, and hence low bandwidth, it may choose to make new spectrum available on which a relay network is formed. This information is broadcast over the cellular control channels so all nodes within the cell receive it simultaneously. Mobile nodes may choose to form/join a relay network operating on the introduced spectrum. Once the relay network is created, nodes forming the relay network relay data to a node acting as a GW which forwards the data to the BS over the cellular interface.

At a later time, if the quality of services from the BS improves or the spectrum in which the relay network is operating is no longer available to the wireless service provider, the relay network is dissolved and nodes return to using the cellular interface with the BS.

In order to limit contention and interference on the relay network, the BS may introduce several spectrum partitions so that multiple relay networks can be formed. Accordingly, each node is equipped with an agile radio so that it may dynamically change frequencies and communication formats to be most suitable based on availability, interference level, or prior arrangement [10, 11].

For simplicity, we focus on a single cell environment in which there is a BS and several mobile nodes. The BS provides a large coverage area (typical radius of 20 km) and mobile nodes within this coverage have a cellular communication link to the BS. Each relay network operates on a single frequency and, in general, the transmission radius of a node on the relay network is very small compared to the cellular coverage. In this paper we assume it is similar to IEEE 802.11b with a transmission range of 115m. Thus, unless the network is highly dense, a relay network generally consists of several isolated groups of mobile nodes as shown in Figure 1.

We use a modified version of AODV [5] as the ad hoc routing protocol to find the path from the mobile nodes to the BS. This is motivated by the fact that the traditional reactive routing protocols are still considered better for small scale networks with a path length of few hops [12]. Similar to AODV, mobile nodes broadcast a route request (*RREQ*) message to find a path to the BS. Intermediate nodes set up the reverse path to the source node and the forward path to the BS.

We have modified AODV so that, like DSR [6], the *RREQ* message contains path information. When an intermediate node forwards a *RREQ*, it appends its identification to the message. Thus, when receiving the *RREQ*, the GW node can learn members of the relay network, and members of specific groups within the relay network. The GW node delivers the *RREQ* to the BS over the cellular interface.

We leverage two optimization features of AODV and DSR to reduce the overhead of forming relay networks. First, a node may *passively learn a route* to a destination, for example if it is part of a longer path to that destination. In this case it will not launch its own *RREQ*. Second, a node that has previously learned a route may *immediately return this route* in response to a request without a further search. Note that this precludes the BS from obtaining a full list of nodes on the relay network during the formation process. We discuss the impact of this in the following subsections.

These two features can greatly reduce the number of messages required to find routes. In order to make the utmost use of the passive route learning, intuitively the furthest node from the BS is the best choice to launch a route request first. This will greatly reduce the number of control messages in the relay network and the load at the GW node because many nodes will passively learn routes.

Motivated by these observations, the five relay network formation algorithms we propose use node location information to varying degrees to schedule control messages during network formation. These schemes are Furthest First (FF), Nearest First (NF), No Wait (NW), Locally Furthest First (LFF), and Region-based LFF (R-LFF).

A relay network is formed in two main phases. In Phase I, a GW node is chosen. While not the focus of our work, we present a basic framework for this phase and discuss how it may be modified for different considerations. In Phase II, the nodes join the relay network by forming a path through one of the GWs to the BS. We consider several methods for this phase to overcome high contention or overload at the GW nodes if all nodes attempt to join the relay network simultaneously.

In the following subsections we discuss the basic GW discovery algorithm and the five different algorithms for applying the routing protocol described above to network formation.

A. Phase I – GW discovery

To select GW nodes, every node periodically broadcasts a neighbor advertisement (*NADV*) message. The *NADV* message contains the identification of the source node and a metric indicative of the quality of its link to the BS. This metric may be a distance from the BS, its received signal strength, or its achievable throughput. We use distance in this paper because it is a reasonable approximation for our purposes and can be easily simulated. It does not impact our results with respect to network formation overhead in any way. When a node receives an *NADV* message from its neighbors, it compares its distance from the BS with its neighbor's. If the node has the shortest distance compared to all neighbors, the node acts as a GW node. The TTL value of *NADV* message is 1.

The GW node performs several functions:

- ***Relaying data***: In order to relay data between the BS and the group, the GW node must be able to use both the cellular and the relay interface simultaneously.
- ***Delivering the group information***: In order to transfer the downlink data to the destination node, the BS needs to keep a list of nodes on the relay network. Thus, the GW delivers the group membership information to the BS during the formation procedure using the information in the *RREQ* messages as described above.
- ***Replying to RREQ***: The GW node is located at the end of relay path. Thus, it should reply to *RREQ* messages received from the group.

B. Phase II – Joining the relay network

In the next five subsections we describe the network formation algorithms. Each formation algorithm defines the order of nodes to initiate a route discovery to the BS. Please refer to Figure 2 for the discussion in the following subsections.

1) NO WAIT (NW)

In this algorithm, all nodes forming a relay network initiate a route discovery to the BS simultaneously. There is no scheduling restriction. This scheme is very simple and easy to implement, but runs the risk of congestion on the relay network during the formation phase, and overloading the GW node with many *RREQ*s during a short time period. To overcome this problem, the remaining algorithms attempt to schedule *RREQ*s for efficient, low latency network formation.

2) FURTHEST FIRST (FF)

To solve the problem of GW overload and congestion during network formation we exploit the ability of nodes to passively learn routes. The motivation is that if a node can passively learn a route, it will not generate a *RREQ.* This optimization reduces both the congestion caused by the potential *RREQ* storm in the relay network, and the load on the GW. From this observation, the most extreme schedule is to have the nodes furthest from the BS launch *RREQ*s first, so that all nodes between this node and the BS will passively learn a route. While this scheme is impractical to implement for reasons discussed below, it provides us with a base line of performance for a strict schedule based on distance.

Because it is impractical for a node to learn the exact location of all other nodes, we assume that the BS acts as a central controller and tracks each node's location in the cell. The BS sorts all nodes in the cell in decreasing order of distance from itself and forces the node furthest from the BS to launch a *RREQ* first. While the furthest node finds a route to the BS, nodes on the path between this node and the GW node passively learn a route also. Nodes that are not on a path from the furthest node to the BS will not be able to passively learn a route. To ensure all nodes learn a route, each is scheduled to launch its own *RREQ* in decreasing order of distance from the BS at every certain time interval, Δt, unless it has passively learned a route.

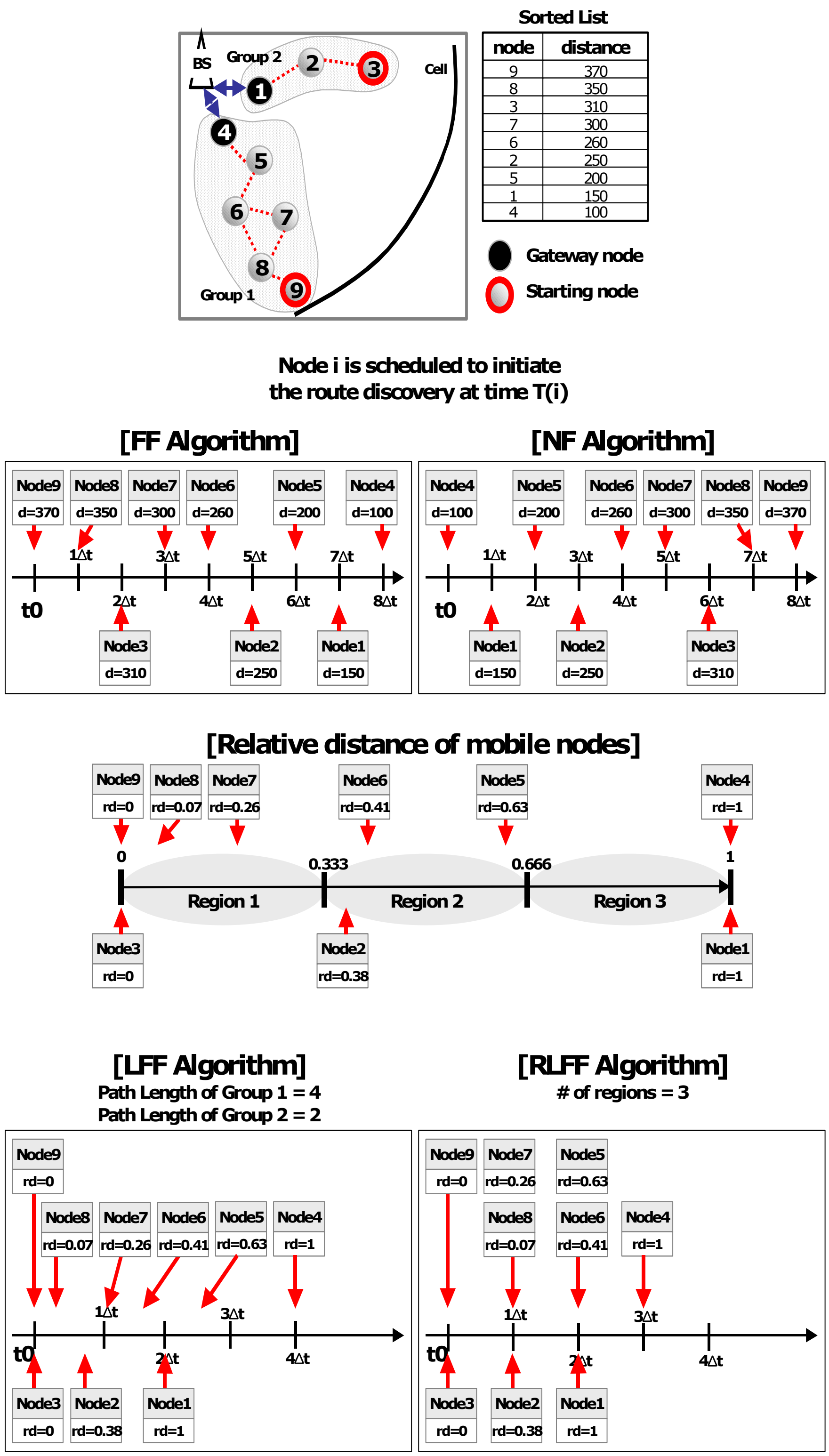
Figure 2 presents the scheduling algorithm of FF scheme and an example. Based on the location information of the nodes, node 9 (furthest from the BS) is scheduled to send out its *RREQ* at $t0$; the next furthest node (node 8) is scheduled at $t0+\Delta t$, node 3 is scheduled at $t0+2\Delta t$, and so on. As shown Figure 3, after node 9 launches its *RREQ* at time $t0$, nodes 5, 6, and 8 may passively learn a route to the BS. Thus, if they receive a route reply (*RREP)* before $t0+6\Delta t$, $t0+4\Delta t$, $t0+\Delta t$, respectively, they will not send out their own *RREQ*s and the time slots assigned to the nodes are wasted.

Note in Figure 2 that two isolated groups exist. The second group is not within range of any nodes in group 1, and hence does not overhear the RREQ broadcasts initiated by node 9. In this group, node 3 will launch a *RREQ* at $t0+2\Delta t$.

```
GW_Discovery( ) {
   Di = Distance from the BS of node i ;
   TTL(NADV) = 1;
   Sending out NADV;
   /* GW node discovery */
   Receiving NADV from neighbors;
   D(Nk) = the distance of neighbor k;
   If (Di == min(Di, D(Nk))) for all neighbors k,
   then node i acts as a GW node;
   /* Starting node discovery */
   If (Di == max(Di, D(Nk))) for all neighbors k,
   then node i acts as a starting node;
}
```

```
Relay_Network_Formation( ) {
   Receiving the broadcast from the BS;
   GW_Discovery();
   t0 = finishing time of GW discovery;
   i = node id;
   d(i) = node i's distance from the BS;
   Node i is scheduled to initiate the route discovery
   at time T(i);

   switch(formation_algorithm) {
   case No_Wait (NW) :
       T(i) = t0;   /* immediate starting */
       break;
   case Furthest_one_First (FF) :
       SD(i) = the index of the node i in the sorted node
               list in decreasing order of distance;
       T(i) = t0 + (SD(i) * Δt);
       break;
   case Nearest_one_First (NF) :
       SI(i) = the index of the node i in the sorted node
             list in increasing order of distance;
       T(i) = t0 + (SI(i) * Δt);
       break;
   case Locally_Furthest_one_First (LFF) :
       If (node i is starting node)
       then
          T(i) = t0;  /* immediate starting */
       else {
          ReceivingGroupInformationfromtheBS( );
          PL = path length (hop count);
          d(G) = the distance of GW node;
          d(S) = the distance of starting node;
          rd(i) = relative distance of node i on the path
                = 1 - (d(i) - d(G)) / (d(S) - d(G))
            T(i) = t0 + (rd(i) * PL * Δt );
       }
       break;
   case Region-based LFF (R-LFF) :
        If (node i is starting node)
        then
           T(i) = t0;  /* immediate starting */
        else {
           ReceivingGroupInformationfromtheBS( );
           d(G) = the distance of GW node;
           d(S) = the distance of starting node;
           rd(i) = relative distance of node i on the path
                 = 1 - (d(i) - d(G)) / (d(S) - d(G))
           X = total # of regions;
           RG(i) = the region of node i
                 = { j | (j-1)/X < rd(i) <= j/X ,  j = 1 .. X }
             T(i) = t0 + (RG(i) * Δt );
        }
   }
```



node	distance
9	370
8	350
3	310
7	300
6	260
2	250
5	200
1	150
4	100

Fig. 2. Algorithm and example of relay network formation

There are two major drawbacks with this algorithm. First, the BS must know the distance to all nodes, which is not practical. Second, since it is impossible to anticipate in advance which nodes may passively learn a route to the BS fixed scheduling must be used. Thus, if a node passively learns a route before its scheduled time, as with nodes 5, 6, and 8 in the example of Figure 2, the time slot is wasted. Thus, FF introduces unnecessary latency into the network formation process.

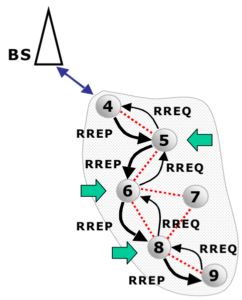


Fig. 3. Passively learning a route to the BS

3) NEAREST FIRST (NF)

When a node already having a forward path to the BS receives a *RREQ*, it immediately returns a *RREP* to the source node. In so doing, the majority of the *RREQ* traffic is replied to without being forwarded to the GW. Therefore, having the node nearest to the BS initiate a *RREQ* first will also lower load on the network. As with FF, this scheme requires the BS to act as a central controller and sort all nodes in the cell in increasing order of distance from the BS.

Figure 2 shows the algorithm and example of NF scheme. As shown in Figure 4, if node 6 has a route to the BS before receiving the *RREQ* initiated by node 9 (at $t0+8\Delta t$ as shown in Figure 2), it returns a *RREP* immediately.

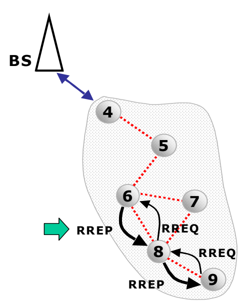


Fig. 4. Immediate response

In addition to requiring a centralized controller, this algorithm has the additional drawback that almost all nodes must launch a *RREQ*, at least to reach the node preceding them to the BS.

4) LOCALLY FURTHEST FIRST (LFF)

FF and NF reduce the number of routing messages flooded in the network. However, mobile nodes may experience long latency due to the strict, sequential scheduling.

To receive the benefits of FF and NF, and the potential low latency of NW through parallelism, we propose the Locally Furthest First (LFF) algorithm. Without any centralized controller, this algorithm allows each node to make a schedule for its own route discovery based on its distance and the group information broadcast by the BS. This algorithm is composed of two steps - starting node discovery and scheduling.

(a) Starting Node Discovery

A relay network generally consists of several isolated groups of mobile nodes as shown in Figure 1. In this algorithm, the starting nodes within each group will launch route discoveries first. The *starting node* is that which has the greatest distance to the BS compared to all neighbors within its transmission range. Thus, they are the nodes that are outmost from the BS within a group. In order to find starting nodes, when a node receives *NADV* message during GW discovery, it compares its distance to the BS with the neighbor's distance. If a node has the greatest distance compared to all neighbors, this node becomes a starting node. Each group has at least one starting node. All starting nodes in the cell initiate a route discovery simultaneously. Thus, several paths will be discovered in parallel.

Figure 2 shows the algorithm and an example of starting node discovery. In this example, nodes 9 and 3 act as the starting node in groups 1 and 2, respectively.

(b) Scheduling

As mentioned in Section III.A, the GW node delivers the group information to the BS when it receives a *RREQ* passed through a new path. In this algorithm, we assume that the BS broadcasts the group information to the mobile nodes in the cell when receiving it.

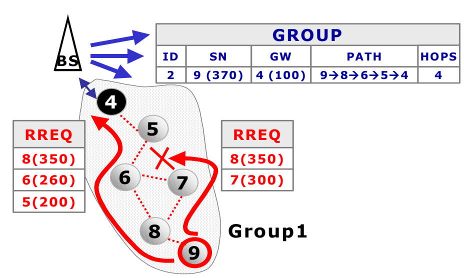


Fig. 5. Group information

As shown in Figure 5, the group information contains the id and distances of the GW node and starting node, the id of intermediate nodes on the path, and the hop count of the path. When receiving the broadcast group information, mobile nodes in the group calculate their relative distance between the GW node and the starting node as shown in the algorithm in Figure 2. The relative distance has a value of the range [0..1]. A smaller value indicates that the node is closer to the starting node. Each node then makes a schedule for its own route discovery with the relative distance.

The group information broadcast by the BS may be incomplete, i.e., it may not include all nodes in the group. For example, in Figure 5, node 6 may discard the *RREQ* forwarded by node 7 since it is a duplicate. In this case, the group information will not include node 7. However, node 7 will have already received a *RREQ* from the starting node 9, so it can learn that node 4 is the GW node matched with the starting node 9 when receiving the broadcast group information. Thus, it can calculate its relative distance between node 4 and 9.

Figure 2 shows the scheduling algorithm of LFF and an example. In this example, since nodes 3 and 9 are starting nodes, they find a route to the BS simultaneously at time t0. Node 6 belongs to the group 1 in which the GW node and the starting node are node 4 and 9, respectively. Node 4's distance is 100 and Node 9's distance is 370. Thus, the relative distance of node 6 is 0.41. The hop count of the path is 4. Thus, node 6 makes a schedule for its route discovery at time $t0+1.64\Delta t$. This scheduled time is proportional to its relative distance. Thus, this algorithm makes use of the passive route learning.

Even though LFF can reduce the formation latency through parallelism, it may still incur high latency with an increase in the node density in the cell. At high node density, more nodes are within transmission range of each other resulting in a fewer number of groups in the cell, and hence, fewer paths discovered in parallel.

5) REGION-BASED LOCALLY FURTHEST FIRST (R-LFF)

In order to increase the parallelism of LFF, we propose the Region-based LFF algorithm (R-LFF). In this algorithm, the path from a starting node to a GW node in each group is divided into pre-defined logical regions. All nodes in the same region initiate a route discovery simultaneously. The time interval Δt is allocated for each region. Thus, nodes in the first region start at $t0+1\Delta t$, the second region at $t0+2\Delta t$, and so on. Since the node's relative distance has a value of the range, [0..1], and indicates the node's relative location on the path, the region of a node can be determined based on the node's relative distance as shown in the algorithm in Figure 2.

In Figure 2, we assume that we have three pre-defined regions. Since node 9 and 3 are starting nodes, they start at time *t0*. Nodes 8, 7 belong to the first region and start at time $t0+1\Delta t$. Nodes 6, 5, and 2 belong to the second region and start at time $t0+2\Delta t$. This example shows that, unlike LFF, all nodes can start their own route discovery within $t0+3\Delta t$ in R-LFF.

C. TRANSIENT BEHAVIOR

As mentioned above, in order to transfer the downlink data to the destination node, the BS needs to keep a list of nodes on the relay network. Thus, the GW node delivers the group membership information to the BS when it receives a *RREQ* passed through a new path. Because some intermediate nodes which have a route to the BS may immediately return a *RREP* to the source node, some *RREQ*s do not reach the GW node. Thus, group information sent by the GW may be incomplete.

For example, as shown in Figure 4, when node 6 receives a *RREQ* sent from node 9, it will reply to the *RREQ* if it has a route to the BS without a further forwarding. Thus, the BS will not receive notification that node 9 has joined the relay network during the formation procedure. In this case the BS will continue to use its cellular interface to communicate with node 9 in the downlink until it receives data from the mobile node via the relay network on the uplink. At this time the BS will have a record of the node being in the relay network group served by a GW and transfer the next downlink data through the relay network.

IV. PERFORMANCE EVALUATION

In this evaluation we focus on the performance of the network formation algorithms and some basic metrics that indicate the quality of the relay networks that are formed. We simulated these formation algorithms in ns-2 version 2.1b9a.

A. PERFORMANCE METRICS

We use five metrics to evaluate the performance of our formation schemes: formation latency, signaling traffic generated, load at the GW nodes during network formation, average path length, and the amount of link sharing.

The first three metrics indicate the overhead of the relay network formation. *Signaling traffic* (M) is the total number of routing messages received by all mobile nodes forming the relay network. It indicates the congestion of the network. *Formation latency* (L) is the time elapsed between the first *RREQ* and all nodes having a route to the BS. *Gateway load* (G) is the total number of routing messages received by all GW nodes. It shows the processing load at GW nodes, and it is proportional to the traffic intensity of the cellular interface between the BS and the GW nodes.

We define the weighted overall overhead of each scheme as follows:

RM(s) = relative signaling traffic of scheme s

$$= \frac{M(s)}{\max\{M(FF), M(NF), M(NW), M(LFF), M(R-LFF)\}}$$

RL(s) = relative latency of scheme s

$$= \frac{L(s)}{\max\{L(FF), L(NF), L(NW), L(LFF), L(R-LFF)\}}$$

RG(s) = relative load at GW nodes of scheme s

$$= \frac{G(s)}{\max\{G(FF), G(NF), G(NW), G(LFF), G(R-LFF)\}}$$

WO(s) = weighted overall overhead of scheme s

$$= (\alpha \times RM(s)) + (\beta \times RL(s)) + (\gamma \times RG(s))$$

, where α, β, γ = weight of metrics, α + β + γ = 1

The remaining two metrics indicate the efficiency of the resulting relay networks. *Average path length* is the average number of hops on the path established between each mobile node in the relay network and the BS. Previous work [14][15] has shown that TCP throughput decreases rapidly as the number of hops increases in a multi-hop wireless network based on 802.11. This is because input and output links of a node must contend with each other. Also, multi-hop 802.11 networks are inherently unfair due to the capture effect. Thus, while path length is not a direct measure of relay network performance, it is a good indicator of the expected TCP throughput.

Link sharing is a measure of traffic aggregation in the relay network. By aggregating many flows onto links, several benefits result. First, the number of competing, and potentially interfering links, is reduced. This will allow better performance from MAC layer protocols such as those used in 802.11 networks that must resolve contention across links. Second, if different frequencies are being assigned to each link, as the number of flows sharing links increases, fewer frequencies will be required in a relay network to achieve orthogonality between links resulting in more efficient spectrum usage. Creating networks with orthogonal links will greatly improve network throughout by eliminating the need for a contention resolution protocol. Finally, more efficient scheduling algorithms may be deployed.

We define *Link Sharing* (S) as the degree of sharing in each node of the relay network normalized against the best case for each size network. This provides a measure of the interfering links present in the relay network. Figure 6 shows a simple example of the sharing. In this example, the relay network consists of 4 nodes including a GW node. Two examples of the resulting relay network are shown in Figure 6.

In case 1, all three nodes have a direct link to the GW. These links are interfering and require a MAC protocol to resolve contention between them. In this case, three frequencies would be required to eliminate the contention through frequency assignments. In case 2, all nodes join a single path to the GW node forming a straight line. Only the adjacent nodes compete for the transmission media, and only two frequencies are needed to eliminate this contention. Case 2 is the optimal case of the sharing.

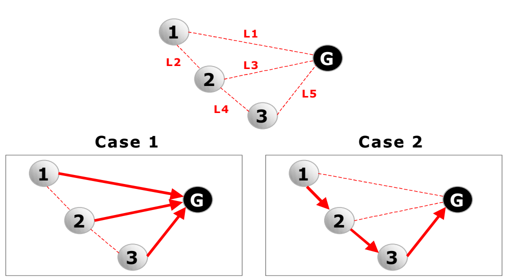


Fig. 6. Example of the sharing

As shown in Figure 6, the number of incoming edges at a node indicates the number of interfering traffic flows passing through the node. We formally define the normalized sharing factor against the optimal case for each size network as

$$S = \frac{\overline{E}_O}{\overline{E}_{net}}$$

where $\overline{E}_O$ is the average number of incoming edges into a node in the optimal sharing case (straight line), and $\overline{E}_{net}$ is the average number of incoming edges into a node in the relay network. Further,

$$\overline{E}_{net} = \frac{\sum_{i=1}^{N} e_i}{N}$$

where e_i is the number of incoming edges into node *i,* and *N* is the number of nodes which have at least one incoming edge. In the optimal case, the average number of incoming edges, $\overline{E}_O$, is 1. Thus, the normalized sharing factor is reduced to

$$S = \frac{N}{\sum_{i=1}^{N} e_i}.$$

In Figure 6, the sharing factor of case 1 is 1 / (3 / 1) = 0.333 and the sharing factor of case 2 is 1 / (3 / 3) = 1.

The main drawback of link sharing is that a small number of nodes may be burdened with a high forwarding load. In these cases paths may be periodically reformed to balance load.

B. EVALUATION

Table 1 summarizes the simulation parameters. Each simulation is run for 30 seconds and each data point in the figure is the average over 100 different topologies. In this simulation, we assume that the mobile nodes joining the relay network use an 802.11-like protocol for control channels during network formation.

Variable	Value
Air interface range	115m
MAC used during network formation	802.11
Cell Size (BS at center)	886m x 886m
Nodes per cell	1➔100
Δt	20 msec
Regions for R-LFF	5

Table 1. Simulation Parameters

Figure 7 shows the latency of each formation scheme. Due to the strict scheduling, NF and FF always have high latency. In LFF, mobile nodes within a group are strictly scheduled depending on their distance, even if each group finds a route to the BS in parallel. Thus, LFF has high latency with increased node density since the number of nodes within a group also increases. Due to the increased parallelism, R-LFF has significantly lower latency than LFF. Despite having the highest parallelism, NW has longer latency than R-LFF. In NW, nodes suffer from severe collisions during formation resulting in the increase in latency.

Figure 8 shows that FF has the lowest signaling traffic and LFF has lower signaling traffic than NF and NW, leading us to conclude that having nodes farthest from the BS initiate a route discovery first is a useful approach. At high node density, R-LFF may have high signaling traffic because many more nodes belong to a common region and find a route in parallel.

Figure 9 shows the load at the GW node for each scheme. NW has the highest load because all nodes launch *RREQ*s close together, so in almost all cases, the *RREQ*s propagate to the GW node. With high node density, R-LFF also has somewhat high GW load because it increases the parallelism of the path search. In general, the schemes with strict scheduling are more effective at passively learning routes, and hence cause less overhead at the GW nodes.

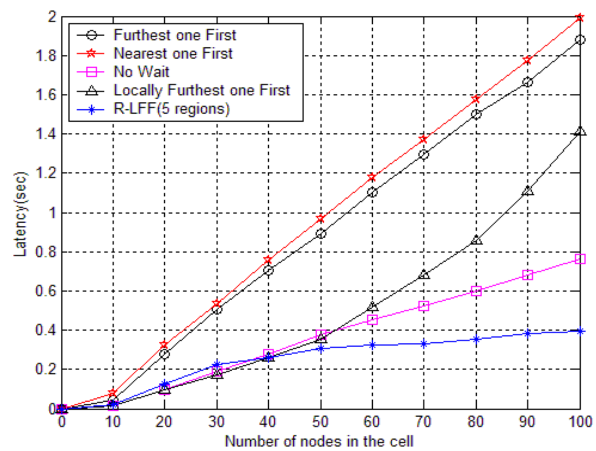


Fig. 7. Formation latency

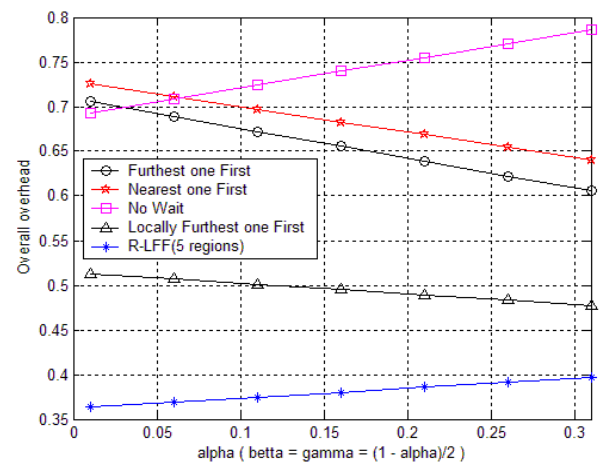


Fig. 10. Overall overhead ($\beta = \gamma = (1-\alpha)/2$)

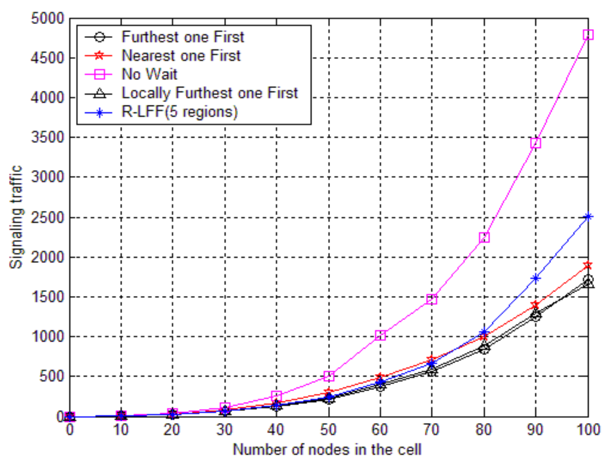


Fig. 8. Signaling traffic

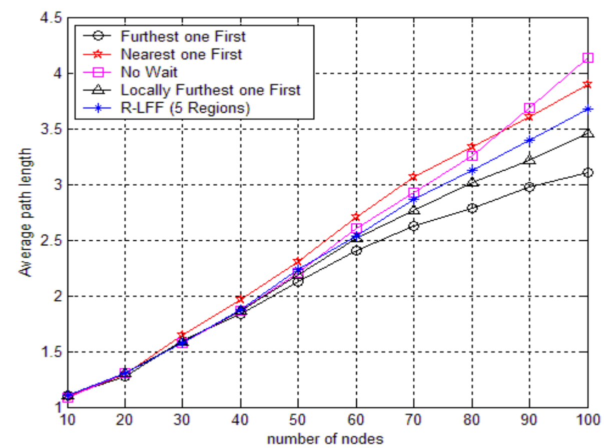


Fig. 11. Average path length

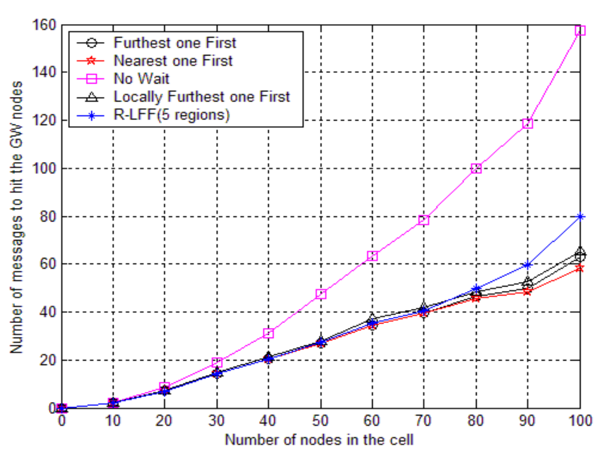


Fig. 9. Load at GW nodes

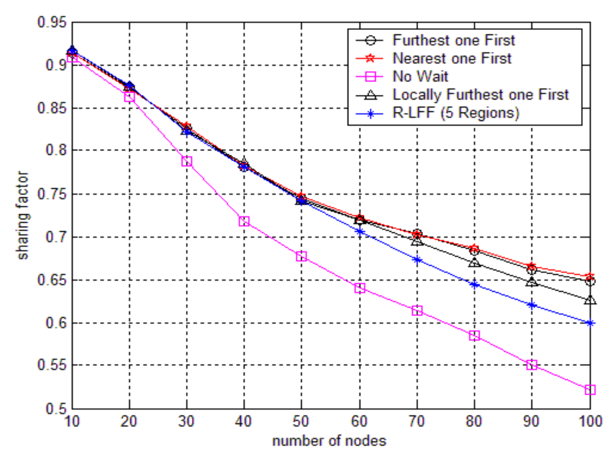


Fig. 12. Sharing factor

When reviewing Figures 7-9 it is clear that all schemes incur trade-offs. NF has the highest latency, but the fewest messages to hit the GW nodes. Even though NW has low latency, the signaling traffic and the load at GW node during network formation is about three times of that when using FF and NF. LFF has good performance in terms of signaling traffic and load at the GW node, but high latency at high node density. Conversely, R-LFF has lower latency than LFF, but high signaling traffic and GW load.

Figure 10 compares the weighted overhead, *WO*, of each scheme as defined in Section IV.A. In this figure, β and γ are set to be equal, meaning that low latency and low load on the GW are of equal importance. We consider latency as highly important because it reflects the disruption in user traffic. GW load is also important because if GW nodes are flooded with *RREQ* messages and have finite buffers, *RREQ*s may be lost. (In our simulations, the buffers in the GW nodes were large). α is varied from 0.01 to 0.33. Thus we evaluate the overhead over the range of when the total message load is considered highly unimportant ($\alpha = 0.01$), through the case in which all three metrics are of equal importance ($\alpha = \beta = \gamma = 0.33$). As shown, R-LFF is the best overall performing algorithm in terms of formation overhead, followed by LFF. NW is the worst performing algorithm except for cases in which total message load is unimportant.

Figure 11 shows that with a small number of mobile nodes in a cell, NW results in the shortest average path length, but the average path length of FF is shortest as the number of mobile nodes increases. It also indicates that FF-based algorithms will generally result in smaller path lengths and hence higher TCP throughput in the relay network.

Figure 12 shows the sharing factor achieved with each method for different size networks. The upper bound is equal to 1. This result shows that by making the utmost use of the passive route learning and the immediate responding, FF and NF have the greatest link sharing. Due to the increased number of flooded *RREQ* messages, NW has the lowest link sharing. This indicates that algorithms with strict ordering (NF and FF) achieve the most sharing, and that sharing decreases as parallelism in the formation algorithm increases.

In summary, FF-based schemes can build relay networks with characteristics that will support high TCP throughput. In addition, the resulting relay networks will be efficient in terms of the number of shared links allowing the great resource sharability. These schemes also are the most efficient in terms of overhead making them a good class of algorithms for relay network formation.

V. DISCUSSION

Below we discuss the correctness of the algorithms and the benefits of pre-backoff.

A. CORRECTNESS

It is possible in a relay network that nodes are isolated, that is they are not in range of any node other than the BS. Likewise, it is possible that bidirectional communication between nodes may not exist, leading one node to believe that another node knows of its existence. These types of problems may result in incorrect scheduling in some cases, i.e., nodes going out of order or at the same time, and will manifest as higher relay network formation latency. They will not preclude nodes from communicating because all nodes will schedule a *RREQ* to the BS as long as they are not the GW node, i.e., they can relay through at least one other node. In cases in which a node is isolated, it may still communicate using its 3G interface directly to the BS.

B. PRE-BACKOFF

In order to reduce the initial collisions in the network, some protocols introduce pre-backoff in which each node waits for a random time before transmitting data. For example, in Bluetooth ad-hoc networks [13], several slave nodes can simultaneously receive a message from a master node. To avoid severe collisions, each slave node chooses a random back-off interval, [0 .. 1023 time slots] before responding the message. DSR [6] uses a similar method to prevent all neighbors from responding to a *RREQ* simultaneously.

Of the algorithms we present, NW and R-LFF experience high collision rates with high node density. We measured the latency of NW under the same conditions as Figure 7 when using a pre-backoff interval of the range [0..10msec]. The results showed that pre-backoff reduces the latency of NW by 30%. When translated into weighted overhead, LFF and R-LFF still always outperform NW; FF outperforms NW when $\alpha > 0.14$; and NF outperforms NW when $\alpha > 0.175$. Therefore, our results do not fundamentally change.

VI. CONCLUSION

In this paper we analyzed the formation of relay networks for dynamic multi-radio, multi-hop wireless cellular networks. We propose five algorithms for network formation, including determining which nodes are best suited for bridging the relay network and the BS, and the order of nodes to initiate a route discovery for establishing paths from nodes in the relay network to the GW nodes.

We leveraged the existence of a BS to broadcast information to assist in establishing this schedule. However, tight scheduling based on distance leads to sequential route discovery, and long latency. Hybrid schemes that afford some degree of parallelism and distance-based scheduling perform better overall. From the evaluation result, we found that schemes scheduling nodes furthest from the BS to initiate route discovery first make good use of passive route discovery and hence reduce the formation overhead. Moreover, they can build efficient relay networks which supports higher TCP throughput and greater sharing of resources. This makes this class of algorithm attractive for forming relay networks.

REFERENCES

[1] W. Hongyi, Q. Chunming, S. De, and O. Tonguz. Integrated cellular and ad hoc relaying systems: ICAR. IEEE Journal on Selected Areas of Communications, 19(10):2105–2115, 2001.

[2] G.M. Aggelou and R. Tafazolli. On the relaying capability of next generation GSM cellular networks. IEEE Wireless Communications, 8(1):40–47, 2001.

[3] Jie (Jay) Zhou and Yang Richard Yang. Parcels: Pervasive Ad-hoc Relaying for Cellular Systems. In Proceedings of Med-Hoc-Net, Sardegna, Italy, September 2002

[4] H. Luo, R. Ramjee, P. Sinha, L. Li, and S Lu. UCAN: a unified cellular and ad-hoc network architecture. ACM MOBICOM, pages 353–367, 2003.

[5] Charles E. Perkins and E.M. Royer, "Ad-Hoc On Demand Distance Vector Routing," IEEE WMCSA'99, New Orleans, LA, February 1999.

[6] David B. Johnson and David A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

[7] Gy. Miklos, A. Racz, Z. Turanyi, A. Valko, and P. Johansson. Performance Aspects of Bluetooth Scatternet Formation. Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing, pages 147-148, 2000

[8] T. Salodinis, P. Bhagwat, and L. Tassiulas. Proximity Awareness and Fast Connection Establishment in Bluetooth. Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing, pages 141-142, 2000

[9] G. Tan, A. Miu, J. Guttag, and H. Balakrishnan. Forming Scatternets from Bluetooth Personal Area Networks. MIT Laboratory for Computer Science, MIT-LCS-TR-826, 2001

[10] J. Glossner, D. Iancu, E. Hokenek, and M. Moudgill. A software-defined communications baseband design. IEEE Communications Magazine, 41(1):120–128, January 2003.

[11] J. H. Reed. Software Radio: A Modern approach to radio engineering. Prentice-Hall, 2002.

[12] Vinod Namboodiri, Manish Agarwal, and Lixin Gao. A study on the feasibility of mobile gateways for vehicular ad-hoc networks. Proceedings of VANET, October 2004.

[13] The Official Bluetooth Specification - Version 1.2 Core Document Number: MISC BLUETOOTH CORE Miscellaneous (Undefined and/or multiple publishers), 2003

[14] G. Holland and N. H. Vaidya. Analysis of TCP Performance over Mobile Ad Hoc Networks. Fifth Annual International Conference on Mobile Computing and Networking (MOBICOM), 1999.

[15] M. Gerla, K. Tang and R. Bagrodia. TCP performance in wireless multi-hop networks. Proceedings of IEEE WMCSA 99, February 1999.